# Privacy-Friendly Collaboration for Cyber Threat Mitigation

Julien Freudiger
PARC, a Xerox Company
Palo Alto, CA

Emiliano De Cristofaro
University College London
London, UK

Alexander Brito
PARC, a Xerox Company
Palo Alto, CA



## ABSTRACT

Sharing of security data across organizational boundaries has often been advocated as a promising way to enhance cyber threat mitigation. However, collaborative security faces a number of important challenges, including privacy, trust, and liability concerns with the potential disclosure of sensitive data. In this paper, we focus on data sharing for predictive blacklisting, i.e., forecasting attack sources based on past attack information. We propose a novel privacy-enhanced data sharing approach in which organizations estimate collaboration benefits without disclosing their datasets, organize into coalitions of allied organizations, and securely share data within these coalitions. We study how different partner selection strategies affect prediction accuracy by experimenting on a real-world dataset of 2 billion IP addresses and observe up to a 105% prediction improvement.

## 1. INTRODUCTION

Despite the media hype, collaborative security approaches are seldom implemented as they raise several important challenges. Security data, such as firewall logs or attack intelligence, might expose confidential and/or sensitive information, challenge corporations' competitiveness, or even reveal negligence.

As a consequence, previous work proposed to sanitize data prior to sharing [30,37,48]. However, this makes data less useful [29] and still prone to de-anonymization [8]. One alternative is to let entities contribute encrypted data to a semi-trusted central repository that obliviously aggregates contributions [3], or use distributed data aggregation protocols based on secure multi-party computation [6]. While aggregation can help compute traffic statistics, it only identifies most prolific attack sources and yields global models. As shown in [43,51], however, generic attack models miss a significant number of attacks, especially when attack sources choose targets strategically and focus on a few known vulnerable networks. In theory, Fully Homomorphic Encryption (FHE) [18] could be used to compute personalized recommendations, however, FHE is still far from being practical and it remains unclear whether complex machine learning algorithms needed for the prediction could effectively run over encrypted data.

**Intuition.** This paper explores a novel approach to collaborative threat mitigation where organizations find suitable collaboration partners in a distributed and privacy-preserving way, and organize into coalitions prior to sharing. This way, sharing takes place within groups of related victims, i.e., that share relevant sources of information. In our model, parties first identify a set of *potential* partners from a larger pool of organizations, e.g., corporations in the same sector, and then select the *best* partners. In practice, this can be repeated over time to ensure relevant and near real-time protection. We introduce the *Sharing is Caring (SIC)* framework, which supports two types of algorithms: one for estimating the benefits of sharing in a privacy-preserving way (i.e., without disclosing plaintext data), and the other for sharing agreed-upon datasets with selected partners, e.g., only common attacks.

We focus on data sharing for predictive blacklisting, namely forecasting attack sources based on logs generated by different organizations' firewalls and/or intrusion detection systems. As shown in previous work [26,43,51], collaboration improves defense accuracy, as attackers tend to target victims in similar ways.

**Experiments.** One of our main goals is to investigate which collaboration strategies work best, in terms of the resulting improvement in prediction accuracy. To this end, we conduct several experiments on a real-world dataset of 2 billion suspicious IP addresses collected by DShield.org [41] over 2 months. This dataset contains a large variety of contributors, as confirmed by our analysis, which allows us to test the effectiveness of data sharing among diverse groups of victims.

**Main Results.** Our analysis yields several key findings, as we observe that: (1) the more information is available about attackers, the better the prediction, as intuitively expected; (2) different collaboration strategies yield a large spectrum of performances, in fact, with some strategies, sharing does not actually help much; (3) sharing information only about common attackers is almost as useful as sharing everything. This highlights both the importance of selecting the right partners and the usefulness of controlled data sharing.

**Summary of Contributions.** Our work is the first to provide a privacy-enhanced solution for collaborative predictive blacklisting. We demonstrate that data sharing does not have to be an "all-or-nothing" process: by relying on efficient cryptographic protocols for privacy-preserving information sharing, it is possible to share relevant data, and only when beneficial. Compared to prior work, our approach has several advantages: (1) it helps privately identify entities with good partnership potential, (2) it minimizes information disclosure, and (3) it increases speed of malicious activity detection, leading to near real-time mitigation. Our work could also be applied to other security-related applications that benefit from data sharing, such as spam filtering [11], malware detection [20], or DDoS mitigation [35].

**Paper Organization.** The rest of the paper is organized as follows. Next section presents some preliminary notions. Section 3 introduces the Sharing is Caring (SIC) framework, while Section 4 presents a measurement-based analysis of a real-world dataset of security logs. Section 5 covers an experimental evaluation of proposed techniques, followed by related work, in Section 6. The paper concludes with Section 7.

## 2. PRELIMINARIES

This section presents our system assumptions and some relevant background information.

### 2.1 System Model

We assume a network of entities $\mathcal{V} = \{V_i\}_{i=1}^{n}$. Each $V_i$ maintaining a dataset $S_i$ of suspicious events, such as suspicious IP addresses observed by a firewall ⟨IP, time, port⟩. We denote this list of events as $L_i$ (for each entity $V_i$). Hence, $S_i = \{L_i\}$. Each entity $V_i$ aims to predict and block (i.e., blacklist) future attacks.

**Existing Approaches.** Thus far, two main approaches have been used for predictive blacklisting: (1) no collaboration, i.e., each entity $V_i$ independently performs the prediction based only its own dataset $S_i$, or (2) community-based, i.e., each entity $V_i \in \mathcal{V}$ submits its dataset $S_i$ to a central repository, which returns a customized blacklist for $V_i$, also based on all entities' datasets. The latter provides increased accuracy [43,51] but requires entities to reveal their datasets to a central repository.

**Our Novel Model.** We introduce a privacy-friendly collaborative model for predictive blacklisting, whereby entities identify good collaboration partners via pairwise secure computations (without the need for a trusted third-party), and then share data. This way, data sharing takes place in groups of related victims. Each entity performs predictions based not only on its own dataset but also on an augmented dataset that comprises information possibly shared by the counterpart, aiming to improve prediction and, at the same time, avoiding the wholesale disclosure of datasets.

**Threat Model.** We denote with $\mathcal{A} \in \mathcal{V}$ an adversary attempting to learn information about other entities' datasets. (External adversaries are not considered, since their actions can be mitigated via standard network security techniques.) In the worst case, $\mathcal{A}$ may try to collaborate with all other entities and collect available information after each data sharing attempt. $\mathcal{A}$ obtains network traces that allow inference of strategic information. Hence, we aim to protect data confidentiality for each $V_i \in \mathcal{V}$. We assume adversary $\mathcal{A}$ to be semi-honest (or honest-but-curious): $\mathcal{A}$ follows protocols' specifications and does not misrepresent any of its inputs, but, during or after protocol execution, it might attempt to infer additional information about other parties' inputs.

### 2.2 Privacy-preserving Information Sharing

We now review some cryptographic primitives used through the rest of the paper.

**Secure Two-Party Computation (2PC)** allows two parties, on input $x$ and $y$, respectively, to privately compute the output of any public function $f$ over $(x, y)$. Both parties learn nothing beyond what can be inferred from the output of the function. For more details on 2PC refer to [23,49].

**Private Set Intersection (PSI)** allows two parties, a server, on input a set $S$, and a client, on input a set $C$, to interact in such a way that the latter only learns $S \cap C$, and the former learns nothing beyond the size of $C$. State-of-the-art instantiations, include both garbled-circuit based techniques [16,22,36] and specialized protocols [14,15,17,25,28]. In our experiments, we use the PSI construction presented in [14], secure under the One-More-RSA assumption [4] in the Random Oracle Model (ROM), with computational and communication complexities linear in set sizes. Note, however, that one can select any PSI construction, without affecting our design.

**Private Set Intersection Cardinality (PSI-CA)** allows two parties, a server, on input a set $S$, and a client, on input a set $C$, to interact in such a way that the latter only learns $|S \cap C|$, while the former learns nothing beyond $|C|$. PSI-CA is a more "stringent" variant than PSI, as it only reveals the magnitude of the intersection, but not the actual contents. There are several instantiations of PSI-CA [2,13,17,21], and, in our experiments, we use the construction presented in [13], which has linear complexities, with security under the One-More-DH assumption [4] in the Random Oracle Model (ROM). Again, note that any PSI-CA construction can be employed.

**Private Jaccard Similarity (PJS)** allows two parties, a server, on input a set $S$, and a client, on input a set $C$, to interact in such a way that the client only learns $J(S, C) = (|S \cap C|)/(|S \cup C|)$, where $J(C, S)$ denotes the Jaccard Similarity index [24] between sets $S$ and $C$. Blundo et al. [5] slightly relax the above definition and shows how to privately compute the Jaccard Similarity index using only PSI-CA. Since $J(S, C) = (|S \cap C|)/(|S| + |C| - |S \cap C|)$, parties can obtain $J(S, C)$ without disclosing the actual sets' content.

### 2.3 Predictive Blacklisting

As mentioned earlier, we focus on predictive blacklisting, i.e., forecasting future malicious sources based on past attacks.

**Algorithm.** Let $t$ denote the day an attack was reported and $T$ the current time, so $t = 1, 2, ..., T$. We partition $T$ into two windows of consecutive days: a training window, $T_{train}$ and a testing window, $T_{test}$. Prediction algorithms rely on information in the training data, $t \in T_{train}$, to tune their model and validate the predictions for the testing data, $t \in T_{test}$.

The Global Worst Offender List (GWOL) is a basic prediction algorithm that selects top attack sources from $T_{train}$, i.e., highest number of globally reported attacks [51]. Local Worst Offender List (LWOL) is the local version of GWOL and operates on a local network based entirely on its own history [51]. LWOL fails to predict on attackers not previously seen, while GWOL tends to be irrelevant to small victims. Thus, machine learning algorithms were suggested to improve GWOL and LWOL [43,51].

We use the *Exponentially Weighted Moving Average* (EWMA) algorithm, as proposed by Soldo et al. [43], to perform blacklisting prediction. EWMA uses time series aggregation: it consists in aggregating attack events from $T_{train}$ to predict future attacks. Other features one could consider include the historical malicious activity of an IP address, the clustering of IP addresses with similar malicious behavior, and the network centrality of a target. It is out of the scope of this paper to improve on existing prediction algorithms – rather, we focus on how to help organizations identify useful partners in a privacy-preserving way, and how different collaboration strategies perform in comparison to each other.

**Accuracy Metrics.** As commonly done with prediction algorithms, we measure accuracy with *True Positives* (TP), which is the number of predictions that correctly match future events. In the blacklisting scenario, TP correspond to the number of attacks in the blacklist that are correctly predicted.

In practice, sources might not be blacklisted at once and blacklisting algorithms might rely on several observations over time before blacklisting a source, such as the rate at which the source is attacking, the payload of suspicious packets, etc. It is important to distinguish between the *prediction algorithm*, which identifies potential malicious sources and/or creates a watch-list from the *blacklisting algorithm*, which actually blocks sources. Blacklisting algo-

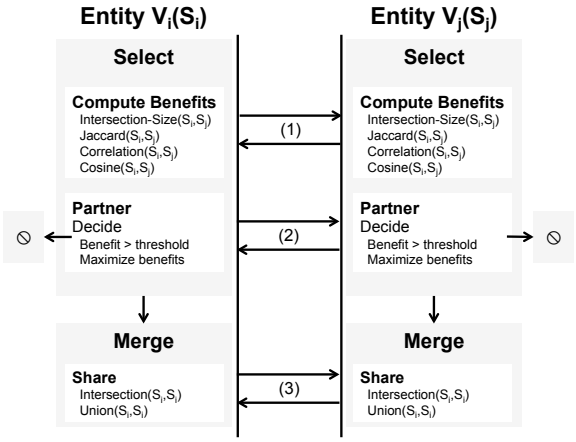



**Figure 1:** Illustration of two entities operating in the SIC framework. (1) Entity $V_i$ starts interacting with entity $V_j$ and they jointly and privately estimate the benefits of collaboration; (2) Entities decide whether or not to partner; (3) Partners decide how to merge their datasets.

rithms are site-specific and need to optimize, among others, false negative and false positive ratios. The prediction algorithm enables the identification of suspicious IP addresses that deserve further scrutiny and improve the effectiveness of blacklisting algorithms.

Therefore, just like prior work [43,51], we focus on measuring the TP of the prediction algorithm, i.e., the ability to identify potential sources of attacks, and do not consider false positives as it is out of the scope of our work.

**Upper Bounds.** A future attack can be predicted if it already appeared in the logs of some victims. Traditional upper-bounds on collaboration algorithms capture this and we use them to evaluate the performance of our collaboration algorithms. The Global Upper Bound $GUB(V_i)$ measures, for every target $V_i$, the number of attackers that are both in the training window *of any victim* and in $V_i$'s testing window. For every $V_i$, we define the Local Upper Bound $LUB(V_i)$, as the number of attackers that are both in $V_i$'s training and testing windows.

# 3. THE SIC FRAMEWORK

## 3.1 Overview

We now describe, in details, the logics behind our intuition for privacy-enhanced collaborative predictive blacklisting by introducing the Sharing Is Caring (SIC) framework. It involves two types of algorithms: one supporting the secure *selection* of collaboration partners, and another for the privacy-preserving *merging* of (i.e., sharing) datasets among partners. As discussed in Section 2, we assume a network of $\{V_i\}_{i=1}^n$ entities and define $S_i$ to be the set of unique IP addresses held by $V_i$: $S_i = \{\text{IP} \in L_i\}$.

A high-level sketch of the SIC framework is presented in Fig. 1. In (1), potential partner entities $V_i$ and $V_j$ estimate the benefits they would receive from sharing their security data with each other. They could do so by securely computing one or multiple metrics. such as size of intersection, Jaccard similarity, Pearson correlation, or cosine similarity between their datasets. In (2), Then, based on the estimated benefits, entities decide whether to partner or not. For instance, $V_i$ and $V_j$ become partners if the expected benefit is above a certain threshold; alternatively, each entity might partner with $k$ other entities that yield the maximum benefits. Finally, in (3), partners merge their datasets, e.g., by only sharing common attacks.

| Benefit Estimation Metric | Operation | Private Protocol |
|---|---|---|
| *Intersection-Size* | $\lvert S_i \cap S_j \rvert$ | PSI-CA [13] |
| *Jaccard* | $\frac{\lvert S_i \cap S_j \rvert}{\lvert S_i \cup S_j \rvert}$ | PJS [5] |
| *Pearson* | $\sum_{l=1}^{N} \frac{(s_{i_l} - \mu_i)(s_{j_l} - \mu_j)}{N\sigma_i\sigma_j}$ | Garbled Circuits [23] |
| *Cosine* | $\frac{\vec{S_i}\vec{S_j}}{\lVert\vec{S_i}\rVert\lVert\vec{S_j}\rVert}$ | Garbled Circuits [23] |

**Table 1:** Metrics for estimating potential benefits of data sharing between $V_i$ and $V_j$, along with corresponding protocols for their secure computation. $\mu_i, \mu_j$ and $\sigma_i, \sigma_j$ denote, resp., mean and standard deviation of $\vec{S_i}$ and $\vec{S_j}$.

## 3.2 Select

Entities select partners by privately evaluating, *in pairwise interactions*, the benefits of sharing their data with each other.

**Supported Metrics.** We consider several similarity metrics for partner selection. Metrics are reported in Table 1, along with the corresponding protocols for their privacy-preserving computation. We consider similarity metrics since previous work [26,51] showed that collaborating with correlated victims works well. Victims are correlated if they are targeted by correlated attacks, i.e., attacks mounted by the same source IP against different networks around the same time. Intuitively, correlation arises from attack trends; in particular, correlated victim sites may be on a single hit list or might be natural targets of a particular exploit (e.g., PHP vulnerability). Then, collaboration helps re-enforce knowledge about an on-going attack and/or learn about an attack before it hits.

**Set-based and Correlation-based Similarity.** We consider two set-based metrics: *Intersection-Size* and *Jaccard*, which measure set similarity and operate on unordered sets. We also consider *Pearson* and *Cosine*, which provide a more refined measure of similarity than set-based metrics, as they also capture statistical relationships. The last two metrics operate on data structures representing attack events, such as a binary vector, e.g., $\vec{S_i} = [s_{i_1}\ s_{i_2} \cdots s_{i_N}]$, of all possible IP addresses with 1-s if an IP attacked at least once and 0-s otherwise. This can make it difficult to compute correlation in practice, as both parties need to agree on the range of IP addresses under consideration to construct vector $\vec{S_i}$. Considering the entire range of IP addresses is not reasonable (i.e., this would require a vector of size 3.7 billion, one entry for each routable IP address). Instead, parties could either agree on a range via 2PC or fetch predefined ranges from a public repository.

In practice, entities could decide to compute any combination of metrics. Note that the list in Table 1 is non-exhaustive and other metrics could be considered, as long as it is possible to efficiently support their privacy-preserving computation.

**Establishing Partnerships.** After assessing the potential benefits of data sharing, entities make an informed decision as to whether or not to collaborate, based, e.g., on:

1. *Threshold:* $V_i$ and $V_j$ partner up if the estimated benefit of sharing is above a certain threshold;
2. *Maximization:* $V_i$ and $V_j$ independently enlist $k$ potential partners to maximize their overall benefits (i.e., $k$ entities with maximum expected benefits);
3. *Hybrid:* $V_i$ and $V_j$ enlist $k$ potential partners to maximize their overall benefits, but also partner with entities for which estimated benefits are above a certain threshold.

| Sharing Strategy | Operation | Private Protocol |
|---|---|---|
| *Intersection* | $S_i \cap S_j$ | PSI [14] |
| *Intersection with Associated Data* | $\{\langle$IP,time,port$\rangle \mid$ IP $\in S_i \cap S_j\}$ | PSI with Data Transfer [14] |
| *Union with Associated Data* | $\{\langle$IP,time,port$\rangle \mid$ IP $\in S_i \cup S_j\}$ | – No Privacy – |

**Table 2:** Strategies for merging datasets among partners $V_i$ and $V_j$, along with corresponding protocols for their secure computation.

While in practice entities could refuse to collaborate with other entities, one could rely on well-known collaboration algorithms that offer stability (e.g., Stable Marriage/Roommate Matching [19]). Without loss of generality, we leave this for future work and assume cooperative parties, i.e., entities systematically accept collaboration requests.

**Symmetry of Benefits.** Some of the protocols used for secure computation of benefits, such as PSI-CA [13] and PJS [5], reveal the output of the protocol to only one party. Without loss of generality, we assume that this party always reports the output to its counterpart. We operate in the semi-honest model, thus parties are assumed not to prematurely abort protocols. Metrics discussed above are *symmetric*, i.e., both parties obtain the same value, and facilitate partner selection as both parties have incentive to select each other.

## 3.3 Merge

After the Select stage, entities are organized into coalitions, i.e., groups of victims that agreed to share data with each other. Entities can now merge their datasets with selected partners.

**Strategies.** Partners could share their datasets in several ways: e.g., they can disclose their whole data or only share which IP addresses they have in common, or transfer all attack events associated to common addresses and/or a selection thereof.

**Privacy-preserving Merging.** Our goal is to ensure that nothing about datasets is disclosed to partners beyond what is agreed. For instance, if partners agree to only share information about attackers they have in common, they should not learn any other information. Possible merging strategies, along with the corresponding privacy-preserving protocols, are reported in Table 2. Again, we assume that the output of the merging protocol is revealed to both parties.

Strategies denoted as *Intersection/Union with Associated Data* mean that parties not only compute and share the intersection (resp., union), but also all events related to items in the resulting set. Obviously, Union with Associated Data does not yield any privacy, as all events are mutually shared. Organizations could also limit the information sharing in time, e.g., by only disclosing data older than a month or of the last week, and previously proposed sanitization techniques [1,30,42] could be used on top of SIC's merging strategies.

## 3.4 Properties

**Privacy.** Our approach guarantees privacy through limited information sharing. Only data explicitly authorized by parties is actually shared. Data sharing occurs by means of secure two-party computation techniques, thus, security follows, provably, from that of underlying cryptographic primitives.

**Authenticity.** Recall that we assume semi-honest adversaries, i.e., entities do not alter their input datasets. If one relaxes this assumption, then it would become possible for a malicious entity to inject fake inputs or manipulate datasets to violate counterpart's privacy. Nonetheless, we argue that assuming honest-but-curious entities is realistic in our model. First, organizations can establish long-lasting relations and reduce the risk of malicious inputs as misbehaving entities will eventually get caught. Also, one could also leverage peer-to-peer techniques to detect malicious behavior [39].

**Incentives and Competitiveness.** Since data exchanges are bi-directional, each party directly benefits from participation and can quantify the contribution of its partners. If collaboration metrics do not indicate high potential, each entity can deny collaboration. That is, the incentive to participate is immediate as benefits can be quantified before establishing partnerships.

**Trust.** SIC relies on data to establish trust automatically. If multiple entities report similar data, then it is likely correct and contributors can be considered as trustworthy. SIC enables entities to estimate each others' datasets and potential collaboration value. This increases awareness of the contribution value and enables automation of trust establishment.

**Speed.** Due to the lack of a central authority and vetting processes, data sharing in SIC is instantaneous, thus, entities can interact as often and as fast as they like.

# 4. THE DSHIELD DATASET

In order to assess the effectiveness of our approach, we should ideally obtain security data from real-world organizations. Such datasets are hard to come by because of their sensitivity. Therefore, we turn to DShield.org [41] and obtain a dataset of firewall and IDS logs mostly contributed by individuals and small organizations. DShield contains data contributors are willing to report, however, as in previous work [43,51], we can assume strong correlation between the amount of reporting and the amount of attacks.

In this section, we show that DShield dataset contains data from a large variety of contributors (in terms of the amount of contributions) and provides a reasonable alternative to experiment with our privacy-enhanced collaborative approach.

## 4.1 The Dataset

We obtained two months' worth of logs from the DShield repository. Each entry in the logs includes a Contributor ID, a source IP address, a target port number, and a timestamp – see Table 3.

| Contributor ID | Source IP | Target port | Timestamp |
|---|---|---|---|
| 44cc551a | 211.144.119.042 | 1433 | 2013-01-01 11:48:36 |

**Table 3:** Example of an entry in the DShield dataset.

The *source* of an attack refers to the attacker and *target* (or contributor) refers to a victim ($V_i$). Note that DShield anonymized the "Contributor ID" field by replacing it with a random yet *unique* string that maps to a single victim. Data obtained from DShield consists of about 2 billion entries, from 800K unique contributors, including more than 16M malicious IP sources, for a total of 170GB. We pre-processed the dataset in order to reduce noise and erroneous entries, following the same methodology adopted by previous work on DShield data [43,51]. We removed approximately 1% of of all entries, which belonged to invalid, non-routable, or unassigned IP addresses, or referred to non-existent port numbers.

## 4.2 Measurements & Observations

We now present a measurement analysis of the DShield dataset, aiming to better understand characteristics of attackers and victims.

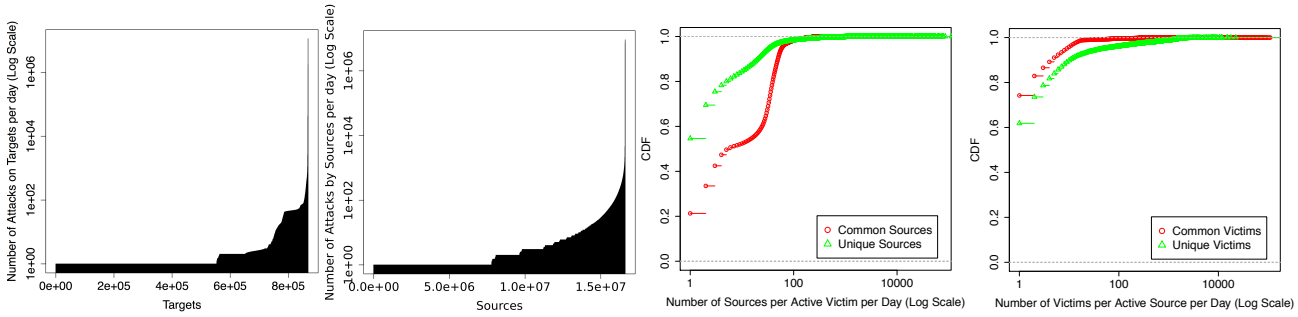



**Figure 2:** Number of attacks per day: (a) on all targets, and (b) by all sources. CDF of the daily number of common and unique: (c) sources per active victims, and (d) victims per active sources. Active refers to the fact that we ignore victims/sources that do not contribute attacks on that specific day to avoid strong bias towards 0.

Overall, our observations are in line with prior work [10,38,43] and highlight how attackers tend to hit victims in a coordinated fashion, thus confirming the potential of collaboration.

**General Statistics.** We observe that 75% of targets contribute less than 10% of the time, while 6% of targets (50,000 targets) contribute daily. We describe, at the end of this section how we filter out targets that seldom contribute. For more details and statistics, we refer to the Appendix.

**Victims' Profile.** Fig. 2(a) shows the number of attacks per day on targets, with mean number of daily attacks on targets of 58.46 and median of 1. We observe three distinct victims' profiles: (1) rarely attacked victims: 87% of targets get less than 10 attacks day, indicating many victims seldom attacked; (2) lightly attacked victims: 11% of victims get 10 to 100 attacks a day; (3) heavily attacked victims: only 2% of targets are under high attack (peaking at 11M a day). In other words, most attacks target few victims.

**Attackers' Profile.** Fig. 2(b) shows the number of victims attacked by each source per day, with mean number of daily attacks of 45.85 and median of 2. We observe that 80% of sources initiate less than 10 attacks a day, i.e., most sources appear stealthy. A small number of sources generates most attacks (up to 10M daily). This indicates two main categories of attackers: stealth and heavy hitters. In our data set, we observe that several of top heavy attackers (more than 20M attacks) come from IP addresses owned by ISPs in the UK.

**Attacks' Characteristics.** Fig. 2(c) shows the CDF of the number of unique sources seen by each active target a day. We focus on active victims: victims that did report an event on that particular day because, as previously discussed, many victims report attacks rarely thus creating a strong bias towards 0 otherwise. The figure contains attackers shared with other targets (common attackers) and attackers unique to a specific victim. 90% of victims are attacked by at most 40 unique sources and 60 shared sources. This shows that, from the victim's perspective, targets observe more shared sources than unique ones. Compared to previous work [26,43], this reinforces the past trend of targets having many common attackers. Fig. 2(d) shows that 90% of sources attack 30 common victims and 60 unique victims. Although attackers share a large number of common victims, they also uniquely attack specific victims. Note that in Fig. 2(c) and Fig. 2(d), we observe again three types of victims and two types of attackers.

**Observations.** A significant proportion of victims (∼70%) contributes a single event overall. After thorough investigation, we find that these *one-time contributors* can be grouped into clusters all reporting the same IP address within close time intervals (often within one second). Many contributors share only one attack event, at the same time, about the same potentially malicious IP address. Similarly, many contributors only contribute one day out of the two months. These contributors correlate with the aforementioned one-time contributors.

We remove victims that do not share much, specifically, we remove victims that (1) share one event overall, and (2) contribute only one day and less than 20 events over the two month (i.e., 10% of mean total contributions per victim 2,263). This data processing maintains properties identified in this section and reduces the number of considered victims from 800,000 to 188,522, corresponding to the removal of about 2 million attacks. This filtering maintains a high diversity of contributors, and seeks to model real-world scenarios (as opposed to focusing on large contributors).

# 5. EXPERIMENTAL EVALUATION

We now present an experimental evaluation of the SIC framework focused on (1) investigating which *select metrics* work best to estimate the benefits of sharing (measured as the resulting improvement in prediction accuracy), and (2) measuring what *merging strategies* (i.e., what data to share) provide the best privacy/accuracy trade-off. To do so, we use the DShield dataset built in Section 4. Experiments involve 188,522 contributing entities, each reporting an average of 2,000 attacks, for a total of 2 billion attacks.

## 5.1 Experimental Setup

Experiments are implemented in R. Source code is available upon request.

**General Parameters.** For the prediction algorithm, we use a one-week window for training ($T_{train} = 7$) and aim to predict attacks for the next day ($T_{test} = 1$). As previously discussed, organizations do not run SIC with all possible other organizations, but focus on a few potential partners. To model this, we take a *sampling* approach: For each iteration, we select 100 victims at random from the set of all 188,522 possible victims and run our select/merge algorithms. We average our results over 100 iterations.

**Select Algorithms.** We analyze how well different collaboration metrics (i.e., select strategies) perform in comparison to each other, where performance is measured in terms of resulting improvement in prediction accuracy.

SIC supports both set-based (*Intersection-Size* and *Jaccard*) and correlation-based (*Pearson* and *Cosine*) metrics. With the former,

the input of each entity $V_i$ is a set of unique attacking IP addresses $S_i$. *Intersection-Size* returns the number of IP addresses attacking both parties, while *Jaccard* is the ratio between the size of set intersection and the size of the union. By contrast, for correlation to work between two entities $V_i$ and $V_j$, they need to agree on the range of IPs captured in $\vec{S_i}$ and $\vec{S_j}$. We assume that both parties know the global list of suspicious IP addresses. In practice, parties can agree on the range via secure computation or fetch known malicious IP address lists from repositories such as DShield.

Metrics are computed pairwise, thus, we obtain a matrix estimating data sharing benefits among all possible pairs. We assume that parties select partners by maximizing their potential benefits in the collaboration matrix. Typically, each party picks the list of partners with the largest potential benefits. W.l.o.g. we consider that the 50 largest collaboration pairs are selected (i.e., only 1% of $100 * 99/2 = 4\ 950$ possible pairs as we consider 100 victims). Such a small number provides a high degree of privacy and takes a conservative stance by limiting the possible improvement in the prediction accuracy. Recall that the goal of our experimental evaluation is to understand *which* metrics work *better*, not to establish the optimal size of collaboration pools.

**Merge Algorithms.** We consider two types of algorithms, *Union with Associated Data* and *Intersection with Associated Data* (see Section 3.3). With the former, partners share all data known by each party prior to current time $t$ and share it with each selected partner. It is a generous strategy that enriches others' datasets rapidly. With the latter, partners only share events from those IP addresses that belong to the intersection (i.e., that attacked both partners) and thus is a more conservative option. This approach can help reinforce knowledge about given adversaries, and thus help better predict attacks.

**Accuracy.** As discussed in Section 2.3, we measure the prediction success by computing the number of True Positives (TP), as in prior work [43,51], i.e., successfully predicted attacks. Specifically, we measure improvement as $I = (\text{TP}_c - \text{TP})/\text{TP}$, where TP is the number of True Positives before collaboration and $\text{TP}_c$ is the number of True Positives after collaboration. We note that improvement can be measured over all entities, or for specific entities. In the following, we give both improvement measures.

## 5.2 Results

**Determining the Value of $\alpha$.** Before testing the performance of select/merge algorithms, we need to identify appropriate $\alpha$ values for the EWMA prediction algorithm by evaluating the performance of the prediction. For small values of $\alpha$, the prediction algorithm aggregates past information uniformly across the training window to craft predictions. In other words, events in the far past have a similar weight to events in the short past and the algorithm has a long memory. On the contrary, with a large $\alpha$, the prediction algorithm focuses on the most recent past events; it has short memory.

Fig. 3(a) shows the evolution of the baseline prediction for different values of $\alpha$, plotting the True Positives (TP) sum of all 100 victims averaged over 100 iterations. Values between $\alpha = 0.5$ and $\alpha = 0.9$ perform best. This can be explained by remembering the "bursty nature" of web attacks, as discussed in Section 4. Prediction algorithms that react fast to the apparition of new attackers perform better. We set $\alpha = 0.9$.

**Visualizing Predictions.** Fig. 3(b) shows a visualization of the prediction. When an attack occurs (blue square), the algorithm systematically predicts an attack (red cross) in the next time slot. Because $\alpha = 0.9$, the last attack event has a larger weight.

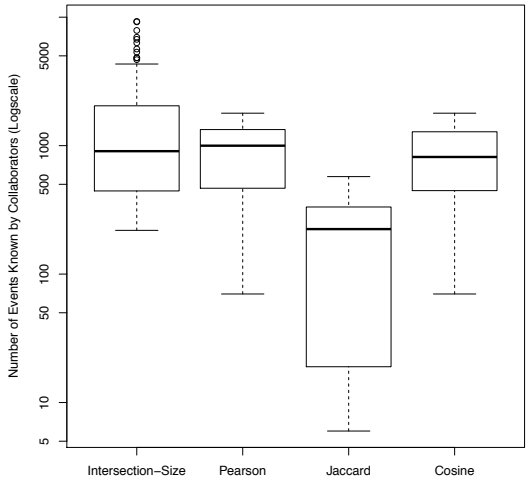



**Figure 5:** Boxplot of number of events known by collaborators given different select metrics. The bottom and top of the box correspond to first and third quartiles. The band inside the box is the second quartile (the median). Outliers are shows with small circles.

**Baseline Prediction.** We verify the effectiveness of the prediction algorithm by correlating the information known prior to collaboration with the ability to predict attacks. We obtain that, as expected, targets that know more about past attacks (large $S_i$), successfully predict more future attacks. We measure correlation $R > 0.9$ on average, which indicates strong correlation. This, once again, suggests that collaboration increases prediction success. We visualize this correlation for a specific simulation in Fig. 4(a).

**Select Strategies.** Fig. 4(b) shows the accuracy of predictions for different select methods over the course of one week, fixing the merge algorithm to *Intersection with Associated Data*, as it provides the strongest privacy protection. We sum the total number of TP for "collaborators" (i.e., entities that do share data) and "non-collaborators" (entities that do not share data, thus performing as in the baseline). We observe that *Intersection-Size* performs best, followed by *Jaccard*, and *Cosine/Pearson*. The overall decrease in sum of True Positives after day 10 is due to the decrease of attacks on those days as discussed in the Appendix (see Fig. 7(a)).

**Improvement Over Baseline.** In Fig. 4(c), we compare the prediction accuracy of upper-bounds, baseline, and collaboration using *Intersection-Size* as the select metric and merging data using *Intersection with Associated Data*. We sum the total number of TP for collaborators selected by the *Intersection-Size* metric. Remind that with the Global Upper Bound (GUB), every victim shares with every other victim and predicts perfectly. With the Local Upper Bound (LUB), organizations do not share anything but still predict perfectly. The accuracy of *Intersection-Size* predictions tends to match LUB, showing that collaboration helps perform as well as a local perfect predictor. Note that prediction performance can be significantly improved (thus, reducing the "gap" with GUB) by enabling more collaboration pairs than the conservative 50 (1% of all pairs) considered in our experiments.

**Effects of Selective Collaboration.** Table 4 summarizes prediction improvements for collaborators given different select metrics, reporting the mean, max, and min improvement, as well as number of collaborators. Correlation-based metrics provide a less significant prediction improvement than set-based metrics. Mean $I$ for *Pearson* and *Cosine* is about 40%. Notably, the *Intersection-Size* has a 105% mean improvement. Also, mean $I$ for *Jaccard* is about

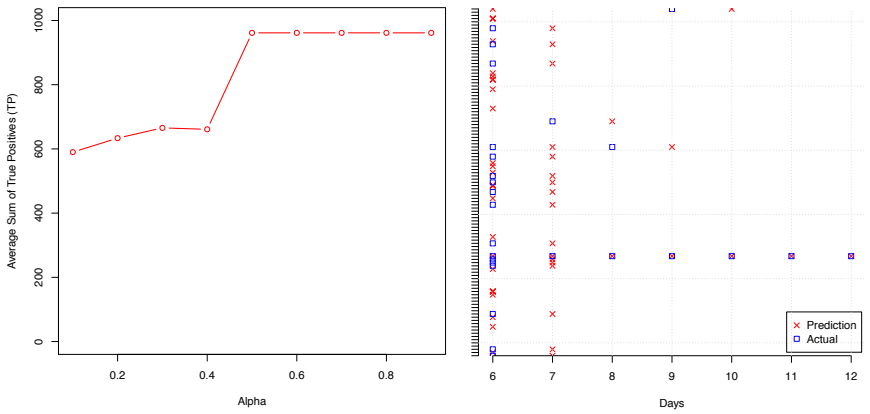



**Figure 3:** Evaluation of baseline prediction (with no collaboration). (a) Number of true positives for different values of prediction algorithm parameter $\alpha$. (b) Visualization of a victim's predictions over time for a series of attackers with $\alpha = 0.9$ on y-axis.

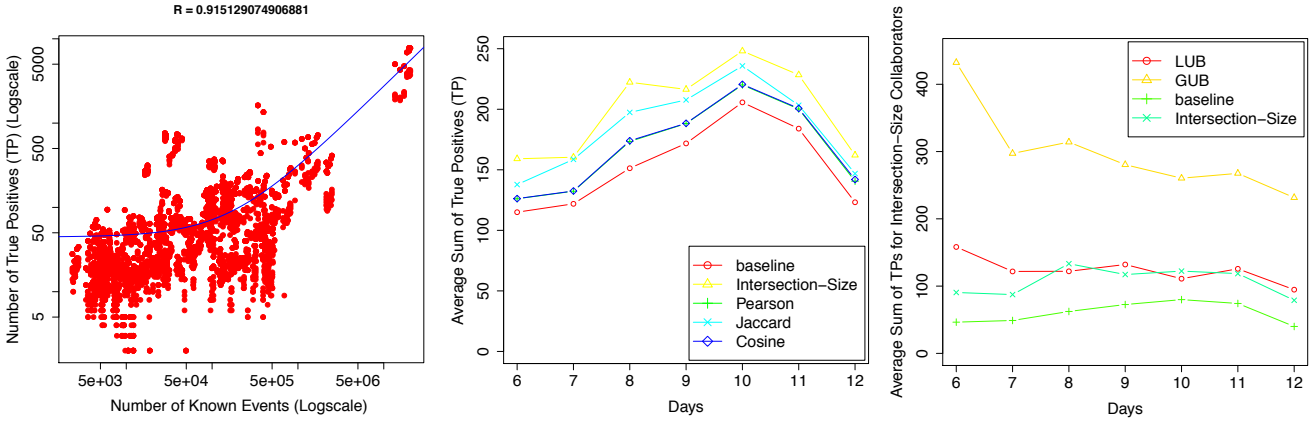



**Figure 4:** Prediction Analysis. (a) Correlation between number of events known by targets, and their ability to predict attacks. The blue curve shows the linear regression (note the log-log scale). (b) Average sum of True Positives over time for different select methods. (c) Average sum of True Positives over time among collaborators selected by *Intersection-Size* including upper bounds (LUB and GUB).

60%. Naturally, the improvement can also be measured for each entity: $I$ for *Intersection-Size* is up to 700%.

Differences between select metrics are due to several reasons. First, metrics that use a normalization factor (i.e., all but *Intersection-Size*) tend to create partnerships of small collaborators. By contrast, *Intersection-Size* leads to better performance because it promotes collaboration with larger victims. To confirm this hypothesis, we measure the set size of collaborators according to different metrics (Fig. 5) and confirm that metrics with a normalization factor tend to pick collaborators that know less. Second, correlation-based metrics tend to select partners that are *too* similar: maximum correlation values are close to 1, whereas maximum *Jaccard* values get to 0.5 only. Although this implies that targets learn to better defend against specific adversaries, it also leads to little acquired knowledge. Third, depending on the select metric, at each time step, victims may partner with previous collaborators, or with new ones. We find that *Intersection-Size*, *Pearson*, and *Cosine* lead to stable groups of collaborators with about 90% reuse over time, whereas *Jaccard* has larger diversity of collaborators over time. This is because about 20% of victims have high *Jaccard* similarity versus only 4% for correlation-based metrics providing a larger pool of potential collaborators. Hence, if *Intersection-Size* helps a few learn a lot, *Jaccard* helps many victims over time.

**Statistical Analysis.** A t-test analysis shows that the mean of the number of events known by collaborators differs significantly ($p < 0.0005$) across all pairs of select metrics but *Cosine* and *Pearson*. If one categorizes collaborators as "large" if they know more than 500 events, and "small" otherwise, and consider *Cosine* and *Pearson* as one (given the t-test result), we obtain a 3X2 table of select metrics and size categories. A $\chi^2$-test shows that categorization differences are statistically significant: *Intersection-Size* tends to select larger collaborators, but also more collaborators than *Pearson/Cosine* (see Table 4). Other metrics tend to select small collaborators. We obtain $\chi^2(2, N = 448) = 191.99, p < 0.0005$, where 2 is the degrees of freedom of the $\chi^2$ estimate, and $N$ is the total number of observations.

**Coalitions.** Recall that, at each time step, entities can decide to partner with a number of other entities. Table 4 shows the mean, standard deviation, and median number of collaborators per party for different collaboration metrics. We observe that with *Jaccard*, entities tend to select less collaborators. Other metrics tend to have similar behavior and have entities to collaborate with about 5 other entities out of 100. This is in line with previous work [26], which

| Select Metric | Improvement | | | Collaborators | | Coalitions | | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Max | Min | Mean | SD | Mean | SD | Med |
| *Int-Size* | 1.05 | 7 | 0 | 19.47 | 2.24 | 5.09 | 4.09 | 4 |
| *Jaccard* | 0.58 | 8 | 0 | 30.17 | 4.44 | 3.16 | 2.74 | 2 |
| *Pearson* | 0.37 | 8 | 0 | 18.08 | 1.40 | 5.20 | 3.15 | 5 |
| *Cosine* | 0.39 | 8 | 0 | 17.98 | 1.29 | 5.26 | 3.14 | 5 |

**Table 4:** Fraction of Prediction Improvements $I$ for Collaborators, number of collaborators, and size of coalitions.

showed the existence of small groups of correlated entities.

We also observe that, after a few days (usually 2), *Intersection-Size*, *Pearson*, and *Cosine* converge to a relatively stable group of collaborators. From one time-step to another, parties continue to collaborate with about 90% of entities they previously collaborated. In other words, coalitions are relatively stable over time. Comparatively, *Jaccard* has a larger diversity of collaborators over time.

**Merge Algorithms.** The next step is to compare the average prediction improvement $I$ for different merge algorithms. As showed in Fig. 6, *Intersection with Associated Data* performs almost as good as *Union with Associated Data* with all select strategies. Actually, it performs better with *Jaccard*. Merging using the union entails sharing more information, thus, one would expect it to always perform better. However, using *Union with Associated Data*, organizations quickly converge to a stable set of collaborators, and obtain a potentially lower diversity of insights over time. With most metrics, the set of collaborators is stable over time anyways, and so union does perform better than intersection. But, as previously discussed, *Jaccard* tends to yield a larger diversity of collaborators over time and thus benefits more from *Intersection with Associated Data* as it re-enforces such diversity of insights.

## 5.3 Performance

We now estimate the operational cost of our techniques and show that it is appreciably low. Specifically, we evaluate the overhead introduced by the privacy protection layer.

Excluding correlation-based metrics (due to lower accuracy improvement), the protocols for privately selecting partners (*Intersection-Size* and *Jaccard*) can be realized based on Private Set Intersection Cardinality (PSI-CA), and we choose the instantiation proposed in [13], which incur computation and communication overhead linear in sets size. Privacy-preserving merging relies on the Private Set Intersection (PSI) with Data Transfer protocol from [14] in order to realize *Intersection with Associated Data*. We implemented protocols from [13] and [14] in C, and conducted experiments on Intel Xeon desktops with 3.10GHz CPU, connected by a 100Mbps Ethernet link. Fig. 2(c) shows that 98% of targets are attacked by about 200 sources. Using sets of size 200, it takes approximately $400ms$ to execute PSI from [14] and $550ms$ for PSI-CA from [13]. Assuming that $n$ organizations contribute to our framework, we have a total of $n-1$ interactions per entity, and a total of $(n-1)n/2$ pairwise executions.

Naturally, it is not reasonable to consider all possible partnerships in a large pool of organizations. Parties first identify a set of *potential* partners, such as organizations within an industry, and then select the *best* partners within. Realistically, we can thus assume $n = 100$. We obtain that the running time amounts to $54s$ for one entity to estimate benefits, using PSI-CA, with all other (99) parties. Following a conservative stance, i.e., assuming that entities select and share with all possible 99 partners, privacy-preserving merging via PSI-DT takes $40s$ (in the worst case scenario).

Pairwise executions can obviously be performed in parallel, at least, between different pairs. Even if we assumed a worst-case scenario, where data sharing occur in a sequential manner across all organizations, the total computation overhead (again, assuming $n = 100$ partners and merging with all partners) would amount to 45 minutes for benefit estimation and 33 minutes for dataset merging, which is still reasonable for computations that are performed, e.g., once a day. Thus, we conclude that overhead introduced by the privacy protection layer is appreciably low and does not impede the deployment and the adoption of our techniques.

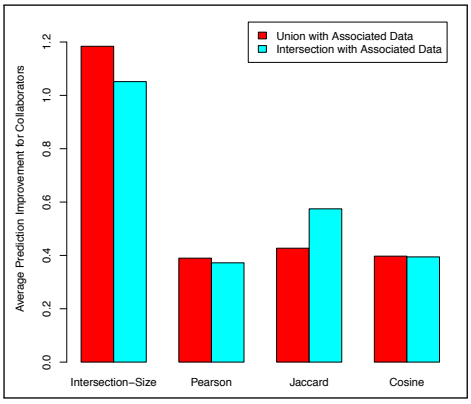



**Figure 6:** Number of True Positives (TP) for two different merge algorithms: *Union/Intersection with Associated Data*.

## 5.4 Summary of Results

**Knowing More Means Predicting More.** Our experiments show that targets that know more tend to successfully predict more attacks. This confirms our hypothesis about the opportunity to collaborate with targets exposed to numerous attacks. However, the simple "more-data-the-better" approach conflicts with privacy, thus, the challenge consists in identifying partners that help most. Choosing partners based on higher values of *Intersection-Size* works best and provides convenient privacy properties since it only discloses information about attackers entities already know of.

**Sometimes Sharing Does Not Help Much.** In some cases, data sharing does not yield significant improvements: we show that differences in similarity definition may lead to significant variations in accuracy. When considering correlation-based similarity between victims' profiles, small contributors are paired together, leading to small overall improvements. By contrast, set-based metrics favor larger contributors and thus yield larger overall improvement.

**Sharing Only Common Attacks Is Almost As Useful As Sharing Everything.** When merging datasets, organizations sharing only information about common attacks (i.e., using *Intersection with Associated Data*) achieve a good trade-off between privacy and utility as the improvement is almost as good as when sharing everything. Intuitively, merging using intersection helps because it reinforces knowledge of a particular attacker, while using union helps victims targeted by varying group of attackers. Thus, victims benefit as much from improving their knowledge of current attackers, as learning about sources that attack them next. In other words, learning information about attackers targeting a victim in the past is similar to learning about attackers that might target a victim in the future.

## 5.5 Limitations

We acknowledge that the DShield dataset used in our experi-

ments might be biased toward small organizations voluntarily reporting data, thus it might not be directly evident how to generalize our results. However, our findings show strong statistical evidence that collaboration metrics affect data sharing performance in interesting ways. Our proposed algorithms and methodology can serve as the basis for further experiments that explore the concept of privacy-enhanced sharing of security-relevant data. Also, as in previous work [43,51], we do not consider false positives but focus on measuring algorithm's TP rate. Nonetheless, as discussed in Section 2.3, this is reasonable as the *prediction* algorithm identifies suspicious IP addresses that deserve further scrutiny and that are subsequently processed by *blacklisting* algorithms, which actually block sources, even though false positives might increase the computational load and complexity of the blacklisting algorithm by providing larger inputs.

Finally, note that this paper does not aim to present a finished product, but to demonstrate the viability and effectiveness of privacy-enhancing technologies on collaborative threat mitigation. While the overhead introduced by our peer-to-peer approach is still non-negligible, it is significantly lower than existing alternatives such as FHE. Also, a few improvements could be explored in future work to improve performance, including parallelization, centralization, and/or sampling.

# 6. RELATED WORK

## 6.1 Collaborative Security Initiatives

**Public Sector.** In 1998, U.S. President Clinton initiated a national program on Critical Infrastructure Protection [7], which promoted collaboration between government and private sector, and created the Financial Sector Information Sharing and Analysis Center (FS-ISAC). In 2003, this was extended to virtual systems and IT infrastructures with the Homeland Security Presidential Directive 7 (HSPD-7), and recently reinforced [34]. In 2013, the US House of Representatives passed the Cyber Intelligence Sharing and Protection Act (CISPA), which met huge opposition as it granted broad immunity to data sharing entities, and took generous views on what data could be shared and with whom. The bill was not voted on by the Senate and the debate is still ongoing with similar proposals [47]. Standardization bodies also push collaborative frameworks and established appropriate data formats (IDMEF, IODEF RFC 5070 [12]), collaboration protocols (the Real-time Inter-network Defense RFC 6545 [32]), and guidelines (ISO 27010, ITU-T SG17).

**Private Sector.** The RedSky Alliance [40] helps security professionals share intelligence after a vetting process for trust establishment. Another example is TF-CSIRT (Task Force of Computer Security Incident Response Teams) [45], which improves coordination among European Community Emergency Response Teams (CERTs). Besides DShield [41], other community-based initiatives focus on sharing and correlating security data. DOMINO (Distributed Overlay for Monitoring InterNet Outbreaks) [50] provides distributed intrusion detection promoting collaboration among nodes. In Europe, the Worldwide Observatory of Malicious Behaviors and Attack Threats (WOMBAT) gathers security related data in real-time. Symantec also introduced a data sharing platform, WINE. Finally, the MITRE Corporation [31] developed file formats (STIX), collaboration protocols (TAXII), and repository formats (CAPEC, MAEC) for structure threat information exchange.

**Barriers to Adoption.** These initiatives have had little success, as pointed out by the Federal Communications Commission's Working Group on Communications Security, Reliability and Interoperability Council's (CSRIC) [9]. Existing solutions rely on manual out-of-band channels to establish *trust*. For instance, the RedSky alliance relies on a long and costly vetting process that requires manual labor to verify the trustworthiness of potential partners. Furthermore, organizations need to reveal their datasets to a centralized third-party and rely on it to for security. Thus, they have limited control over how their data is shared with other participants.

These solutions have a turnover of a few days for RedSky alliance, to a few weeks for ISACs. Feedback is significantly slower than the spread of malware. It is difficult for companies to quantify how much others are contributing, and the lack of transparency discourages contributions.

## 6.2 Collaborative Threat Mitigation

Most of previous works for collaborative predictive blacklisting [26,38,43,51] rely on central repositories and provide no privacy protection. Katti et al. [26] show that correlated attacks, i.e., mounted by same sources against different victims, are prevalent on the Internet. They cluster victims that share common attacks and find that: (1) correlations among victims are persistent over time, and (2) collaboration among victims from correlated attacks improves malicious IP detection time. Pouget at al. [38] also obtain similar results using distributed honeypots for observation of malicious online activities. Then, Zhang et al. [51] experiment with predictive blacklisting, suggesting that victims can predict future attackers, with significantly improved accuracy, based on their logs and those of other similar victims. Soldo et al. [43] also aim to forecast attack sources based on shared attack logs, using an implicit recommendation system and improve on prediction accuracy as well as robustness against poisoning attacks.

## 6.3 Privacy-Preserving Data Sharing

As data sharing raises important confidentiality and privacy concerns, the security community has suggested, in slightly different contexts, to use anonymization or cryptographic techniques to protect privacy. Lincoln et al. [30] suggest sharing sanitized security data for collaborative analysis of security threats. Specifically, they remove, prior to sharing, sensitive data such as IP addresses. Other mechanisms include prefix-preserving anonymization of IP addresses [42,48] and statistical obfuscation [1]. However, inference attacks can de-anonymize network traces [8], and it is quite difficult to maintain data utility [27,29,33,46].

Applebaum et al. [3] introduce privacy-preserving data aggregation protocols geared for anomaly detection. Their approach requires a semi-trusted proxy aggregator and only provides participants with aggregated counts of common data points across multiple entities. Burkhart et al. [6] explore a distributed solution, based on secure multi-party computation and secret sharing, that supports aggregation of security alerts and traffic measurements among peers, e.g., to estimate global traffic volume. These protocols are secure as long as the majority of peers do not collude, assume a reliable infrastructure to distribute key shares, and incur a large number of rounds and high communication overhead.

While aggregation can help compute traffic statistics, it mainly identifies most prolific attack sources and yields global models. However, as shown in [43,51], generic attack models miss a significant number of attacks, especially when attack sources choose targets strategically and focus on a few known vulnerable networks. In theory, Fully Homomorphic Encryption (FHE) [18] could be used to compute personalized recommendations, but FHE is still far from being practical and it is unclear whether complex prediction algorithms could effectively be run over encrypted data.

# 7. CONCLUSION

This paper presented a novel privacy-friendly approach to collaborative threat mitigation. We showed how organizations can quantify expected benefits in a privacy-preserving way (i.e., without disclosing their datasets) before deciding whether or not to collaborate. Based on these benefits, they can then organize into coalitions and decide what/how much to share. We focused on collaborative predictive blacklisting, evaluated our techniques on a real-world dataset, and observed a significant improvement in prediction accuracy (up to 105%, even when only 1% of all possible partners collaborate).

Our analysis showed that some collaboration strategies work better than others. The number of common attacks provides a good estimation of the benefits of sharing, as it drives entities to partner with more knowledgable collaborators. Interestingly enough, only sharing information about common attacks proves to be almost as useful as sharing everything. This suggests that victims benefit as much from improving their knowledge about entities that currently attack them, as from learning about entities that do not attack them now, but might in the future.

We demonstrated the benefits of privacy-preserving information sharing on collaborative threat mitigation and established that data sharing does not have to be an "all-or-nothing" process: by relying on efficient secure computation, it is possible to only share relevant data, and only when beneficial. Privately assessing whether or not, and how, entities should partner up prompts interesting challenges, which our work is really the first to tackle. As part of future work, we intend to study other metrics for partner selection (e.g., dissimilarity) and experiment with other prediction algorithms and incentive mechanisms. We will also explore how to adapt our approach to other collaborative security problems, e.g., spam filtering [11], virus detection [20], or DDoS mitigation [35].

# 8. REFERENCES


[1] E. Adar. User 4xxxxx9: Anonymizing query logs. In *Query Log Analysis Workshop*, 2007.

[2] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *SIGMOD*, 2003.

[3] B. Applebaum, H. Ringberg, M. Freedman, M. Caesar, and J. Rexford. Collaborative, privacy-preserving data aggregation at scale. In *PETS*, 2010.

[4] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3), 2003.

[5] C. Blundo, E. De Cristofaro, and P. Gasti. EsPRESSo: Efficient Privacy-Preserving Evaluation of Sample Set Similarity. *JCS*, 22(3), 2014.

[6] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics. In *Usenix Security*, 2010.

[7] W. J. Clinton. Presidential Decision Directive 63. `http://www.fas.org/irp/offdocs/pdd/pdd-63.html`, 1998.

[8] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, M. K. Reiter, et al. Playing Devil's Advocate: Inferring Sensitive Information from Anonymized Network Traces. In *NDSS*, 2007.

[9] CSRIC Working Group 7. U.S. Anti-Bot Code of Conduct for Internet Service Providers:Barriers and Metrics Considerations, 2013.

[10] M. Dacier, V.-H. Pham, and O. Thonnard. The wombat attack attribution method: some results. In *Information Systems Security*. 2009.

[11] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. P2P-based collaborative spam detection and filtering. In *P2P*, 2004.

[12] R. Danyliw, J. Meijer, and Y. Demchenko. The Incident Object Description Exchange Format. IETF RFC 5070, 2007.

[13] E. De Cristofaro, P. Gasti, and G. Tsudik. Fast and Private Computation of Cardinality of Set Intersection and Union. In *CANS*, 2012.

[14] E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *FC*, 2010.

[15] E. De Cristofaro and G. Tsudik. Experimenting with fast private set intersection. In *TRUST*, 2012.

[16] C. Dong, L. Chen, and Z. Wen. When Private Set Intersection Meets Big Data: An Efficient and Scalable Protocol. In *CCS*, 2013.

[17] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, 2004.

[18] C. Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2009.

[19] D. Gusfield and R. W. Irving. *The stable marriage problem: structure and algorithms*. MIT Press Cambridge, 1989.

[20] B. T. Hailpern, P. K. Malkin, R. J. Schloss, et al. Collaborative server processing of content and meta-information with application to virus checking in a server network, 2001. US Patent 6,275,937.

[21] S. Hohenberger and S. Weis. Honest-verifier private disjointness testing without random oracles. In *PETS*, 2006.

[22] Y. Huang, D. Evans, and J. Katz. Private Set Intersection: Are Garbled Circuits Better than Custom Protocols? In *NDSS*, 2012.

[23] Y. Huang, D. Evans, J. Katz, and L. Malka. Faster secure two-party computation using garbled circuits. In *Usenix Security*, 2011.

[24] P. Jaccard. Etude comparative de la distribution florale dans une portion des Alpes et du Jura, 1901.

[25] S. Jarecki and X. Liu. Fast secure computation of set intersection. In *SCN*, 2010.

[26] S. Katti, B. Krishnamurthy, and D. Katabi. Collaborating against common enemies. In *IMC*, 2005.

[27] E. Kenneally and K. Claffy. Dialing privacy and utility: A proposed data-sharing framework to advance Internet research. *IEEE Security & Privacy*, 8(4), 2010.

[28] L. Kissner and D. Song. Privacy-preserving set operations. In *CRYPTO*, 2005.

[29] K. Lakkaraju and A. Slagell. Evaluating the utility of anonymized network traces for intrusion detection. In *Securecomm*, 2008.

[30] P. Lincoln, P. Porras, and V. Shmatikov. Privacy-preserving sharing and correction of security alerts. In *Usenix Security*, 2004.

[31] MITRE Corporation. STIX. `http://stix.mitre.org/about/documents.html`.

[32] K. Moriarty. Real-time Inter-network Defense (RID). IETF RFC 6545, 2012.

[33] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov. BotGrep: Finding Bots with Structured Graph Analysis. In *Usenix Security*, 2010.

[34] B. Obama. The 2013 State of the Union. http://www.whitehouse.gov/state-of-the-union-2013, 2013.
[35] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson. A framework for a collaborative DDoS defense. In *ACSAC*, 2006.
[36] B. Pinkas, T. Schneider, and M. Zohner. Faster private set intersection based on OT extension. In *Usenix Security*, 2014.
[37] P. Porras and V. Shmatikov. Large-scale collection and sanitization of network security data: risks and challenges. In *Workshop on New security paradigms*, 2006.
[38] F. Pouget, M. Dacier, and V. H. Pham. Vh: Leurre. com: on the advantages of deploying a large scale distributed honeypot platform. In *E-Crime and Computer Conference*, 2005.
[39] M. Raya, P. Papadimitratos, V. Gligor, and J.-P. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *INFOCOM*, 2008.
[40] Red Sky Alliance. http://redskyalliance.org/.
[41] SANS Technology Institute. DShield Data. https://www.dshield.org/.
[42] A. Slagell and W. Yurcik. Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization. In *Securecomm*, 2005.
[43] F. Soldo, A. Le, and A. Markopoulou. Predictive blacklisting as an implicit recommendation system. In *INFOCOM*, 2010.
[44] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, pages 1018–1021, 2010.
[45] TERENA. TF-CSIRT. http://www.terena.org/activities/tf-csirt/, 2013.
[46] Titan Threat Intelligence System. http://www.gtresearchnews.gatech.edu/titan-threat-intelligence-system/, 2013.
[47] US Congress. Cyber Intelligence Sharing and Protection Act. http://beta.congress.gov/bill/113th-congress/house-bill/624, 2013.
[48] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *ICNP*, 2002.
[49] A. Yao. Protocols for secure computations. In *FOCS*, 1982.
[50] V. Yegneswaran, P. Barford, and S. Jha. Global Intrusion Detection in the DOMINO Overlay System. In *NDSS*, 2004.
[51] J. Zhang, P. A. Porras, and J. Ullrich. Highly predictive blacklisting. In *Usenix Security*, 2008.

# APPENDIX

# A. MORE ON THE DSHIELD DATASET

In this appendix, we provide further details about the DShield dataset.

## *General Statistics*

We start by presenting, in Fig. 7(a), the histogram of the number of attacks per day, indicating about 30M daily attacks. We observe a significant increase around day 50 to 100M attacks. Careful analysis reveals that a series of IP addresses start to attack more aggressively around day 50, indicating what might be the beginning of a DoS attack.

Fig. 7(b) then shows the number of unique targets and sources over time. A detailed analysis shows a relatively stable number of sources and targets. This stability in number of attackers confirms that it should be possible to predict attackers' tactics based on past observations. An analysis of attacked ports shows that top 10 attacked ports (with more than 10M hits) are Telnet, HTTP, SSH, DNS, FTP, BGP, Active Directory, and Netbios ports. This shows a clear trend towards misuse of popular web services.

Next, in Fig. 8, we plot the CDF of the fraction of victims that contribute their logs to DShield over the course of two months.

## *Predictability*

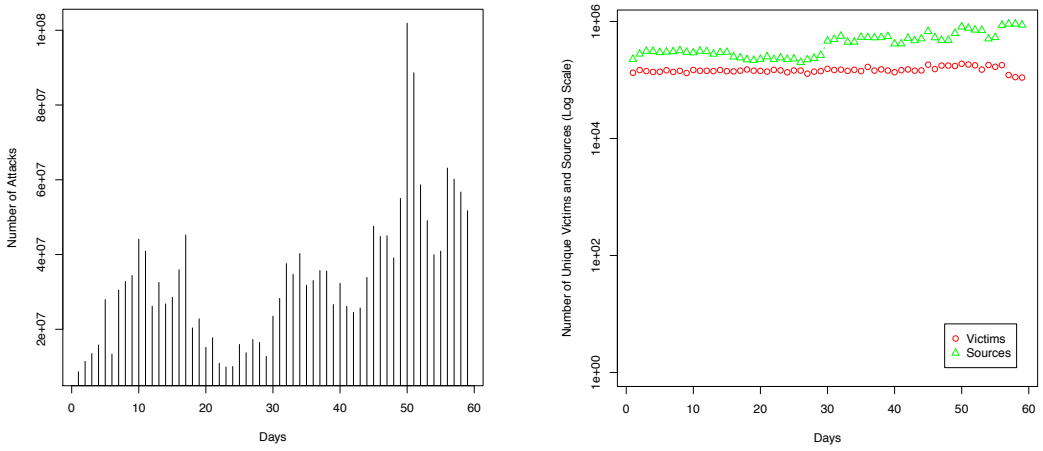
Fig. 9 shows the CDF of the Shannon entropy of the different log entry elements. It helps us visualize the uncertainty about a given IP address, port number or target appearing in the logs, and thus estimate our ability to predict those values. To obtain this figure, we estimate the probability of each victim, source or port being attacked each day. For example, for each port $i$, we compute:

$$\Pr(\text{Port } i \text{ on day } j) = \frac{\text{Attacks on Port } i \text{ on day } j}{\text{Attacks on day } j} \quad (1)$$

We also compute the entropy for each day and aggregate it overall using the CDF. Previous work [44] showed that, following Fano's inequality, entropy correlates with predictability. We observe that ports numbers have the lower entropy distribution, indicating a small set of targeted ports: $80\%$ of attacks target a set of $2^7 = 128$ ports, indicating high predictability. We also observe that victims are more predictable than sources, as $90\%$ of victims lie within a set of $2^{12} = 4096$ victims as compared to $90\%$ of sources being in a list of $2^{14} = 16,384$ sources. Victims' set is thus significantly smaller and more predictable than attackers' set.

## *Intensity*

Fig. 10(a) shows the inter-arrival time of attacks in hours, and Fig. 10(b) shows the inter-arrival time of attacks in seconds. We observe that almost all attacks occur within 3-minute windows. IP addresses and $/24$ subnetworks have similar behavior. In particular, Fig. 10(b) shows that in short time intervals, $85\%$ of $/8$ subnetworks have short attack inter-arrival time indicating the bursty attacks on such networks. Attackers target subnetworks for short time and then disappear.




(a) (b)

**Figure 7:** General DShield characteristics: (a) Histogram of number of attacks per day. (b) Number of unique targets and sources.

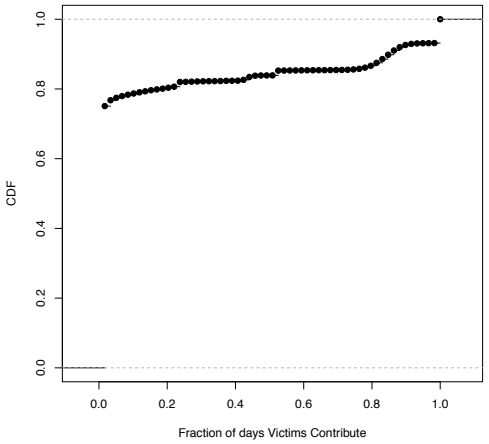



**Figure 8:** Fraction of days each target contributes.

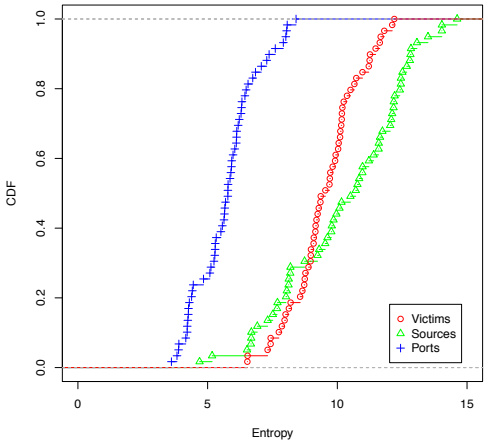
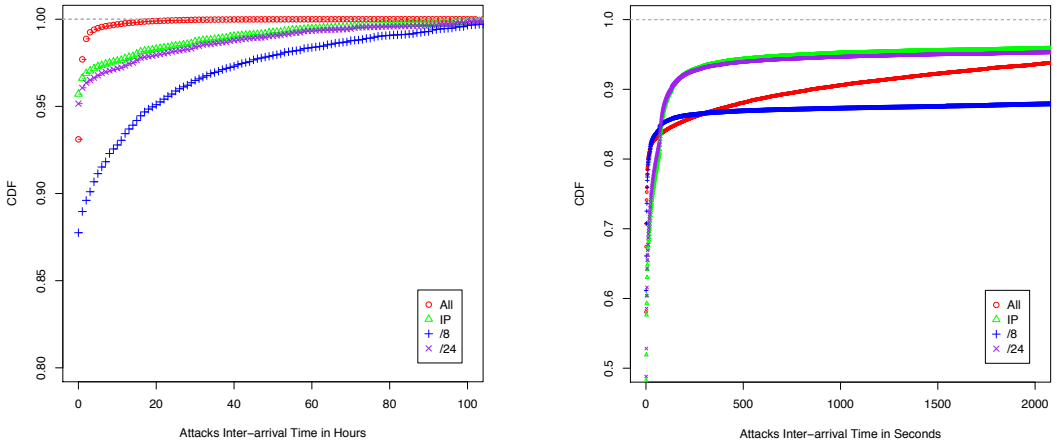



**Figure 9:** CDF of entropy of different attack parameters.




(a) (b)

**Figure 10:** CDF of inter-arrival time of attacks: (a) per hour, and (b) per second. All indicates the inter-arrival time of any attacks, /8 of common /8 subnetworks, /24 of common /24 subnetworks, and IP of the same IP.